

## Задача А. Эстафета

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

В Диманово близится очередной спортивный праздник. Все школы города будут соревноваться в эстафетном забеге на центральном стадионе. Команда каждой школы состоит из трех человек – один девятиклассник, один десятиклассник и один одиннадцатиклассник. Участники команды будут бежать по очереди, каждый по одному этапу эстафеты. Смена участника в команде происходит, когда предыдущий участник завершает бежать свой круг.

Чтобы соревнования получились более конкурентными, организаторы придумали следующие правила для распределения мест после финиша: на каждом этапе забега команда получает число штрафных баллов, равное количеству участников других команд, бежавших этот же этап и финишировавших раньше этой команды. В итоге баллы за все три этапа суммируются, и первое место занимает команда (или команды), которая получила меньше всего штрафных баллов. Второе место – команда (или команды), которая набрала следующее число штрафных баллов, после победителя и т.д. Естественно несколько команд могут получить одинаковое число баллов, тогда они делят одинаковое место, также на каждом этапе некоторые школьники могут финишировать одновременно, в этом случае они получают одинаковое число баллов, согласно правилам.

Учитель физкультуры Василий Петрович первый раз столкнулся с такими правилами на соревнованиях и решил посчитать, на какое место может рассчитывать команда его школы. Сегодня утром на тренировочных забегах он записал времена всех участников. Однако с математикой у него всегда было плохо, поэтому ему требуется помощь в расчетах, какое место может занять его команда.

### Формат входных данных

В первой строке входного файла содержится два целых числа  $N$  и  $I$ , где  $N$  – это количество команд, которые будут принимать участие в забеге,  $I$  – номер команды Василия Петровича ( $1 \leq N \leq 1000$ ,  $1 \leq I \leq N$ ). Все команды пронумерованы числами от 1 до  $N$ .

Следующие  $N$  строк содержат описание участников соответствующей команды. Каждая строка содержит три целых числа  $A$ ,  $B$ ,  $C$  – времена участников команды, за которые они утром пробежали круг стадиона, при этом  $A$  – время того, кто побежит первый этап,  $B$  – время участника второго этапа,  $C$  – время третьего ( $100 \leq A, B, C \leq 200$ ).

### Формат выходных данных

В выходной файл нужно вывести одно целое число – место, которое займет команда Василия Петровича, если участники побегут также, как в утреннем тренировочном забеге.

### Примеры

<code>input.txt</code>	<code>output.txt</code>
2 1 112 107 101 109 105 106	2
3 2 101 100 101 102 100 100 100 101 101	3

### Замечание

В первом примере первая команда получает по одному штрафному баллу на первом и втором круге. К финишу обе команды приходят одновременно, поэтому на третьем этапе обе команды получают нулевое число штрафных баллов. В итоге вторая команда выигрывает, поскольку сумма ее штрафных баллов равна нулю против двух у первой команды.

## Задача В. Стикеры

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

На уроке математики в главной школе города Диманово неожиданно закончился мел. Ученики изобретательно подошли к решению этой проблемы – они взяли набор стикеров, на каждом из них написали по одной цифре, и составили нужные для урока числа.

Прозвенел звонок на перемену, а на доске остались висеть  $N$  чисел. Лучшего математика класса Роя заинтересовало, чему равен остаток от деления на  $M$  суммы чисел на доске. Рой хочет, чтобы этот остаток был равен в точности  $T$ , и естественно, он постарается добиться своего. Для этого, он может взять пару стикеров на доске, и поменять их местами. Его не смущает даже тот факт, что в получающихся числах могут появиться лидирующие нули. Стоит, правда, отметить, что перемена между уроками достаточно короткая, а менять стикеры местами – не быстрый процесс, поэтому Рой может сделать эту операцию последовательно не более двух раз.

Помогите Рою по заданной последовательности чисел определить, сможет ли он, не более чем за две перестановки цифр, получить числа, сумма которых при делении на  $M$  дает остаток  $T$ .

### Формат входных данных

В первой строке входного файла задано три целых числа, разделённых пробелом  $N$ ,  $M$  и  $T$  ( $1 \leq N \leq 10^5$ ,  $1 \leq M \leq 10^9$ ,  $0 \leq T \leq M - 1$ ).

Вторая строка содержит ровно  $N$  целых чисел  $a_1, a_2, \dots, a_N$ , разделённых пробелом – исходные числа, составленные из стикеров ( $1 \leq a_i \leq 10^8$ ). Гарантируется, что в исходной последовательности нет чисел с лидирующими нулями.

### Формат выходных данных

В первую строку выходного файла необходимо вывести число  $k$  – количество действий, которые нужно сделать Рою ( $0 \leq k \leq 2$ ).

Далее нужно вывести  $k$  строк, каждая из которых содержит четыре числа:  $n_1, p_1, n_2, p_2$  ( $n_1 \neq n_2$  или  $p_1 \neq p_2$ ) – номер числа и разряда первого стикера и номер числа и разряда второго стикера, которые нужно поменять местами. Числа нумеруются с единицы, начиная с крайнего слева. Разряды чисел нумеруются с единицы, где 1 – единицы, 2 – десятки и т.д.

Если возможных вариантов ответа несколько, выведите любой из них. Гарантируется, что ответ существует.

### Примеры

input.txt	output.txt
1 200 100 100	0
3 1000 39 123 1 456	2 2 1 3 2 3 1 1 3
1 500 110 101	1 1 1 1 2

## Задача С. Студия звукозаписи

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Меломан Коля работает на студии звукозаписи "Dimanovo Records". Недавно состоялось знаменательное событие в жизни студии – она переехала на новое место. Однако переезд был произведен в достаточно сжатые сроки, и вся аппаратура перевозилась на скорую руку. В результате на пульте диджея перепутались провода, часть из них переплелась между собой, образуя узлы. Теперь совсем непонятно какие провода идут между узлами, какой из них ведет к колонкам, какой к микрофону, какой к ноутбуку и т.д.

Коля задумался, сколько же времени у него займет, чтобы восстановить соответствие проводов входам на пульте. Для каждой проверки он подключает то, или иное устройство к одному из проводов и пробует им воспользоваться. Если устройство заработало, то провод был подключен правильно. Все устройства различны, и каждому устройству соответствует лишь один из проводов. Необходимо посчитать, какое наименьшее количество проверок ему нужно будет сделать, чтобы гарантированно восстановить нужную конфигурацию проводов.

### Формат входных данных

В первой строке входного файла содержится три целых числа  $N$ ,  $M$ ,  $K$ , где  $N$  – количество проводов, подсоединенных к пульту,  $M$  – количество узлов, завязавшихся на проводах, и  $K$  – количество строк, описывающих перепутавшиеся провода ( $1 \leq N \leq K \leq 100000$ ,  $0 \leq M \leq 100000$ ). Концы проводов, которые уже подключены к пульту диджея, нумеруются числами от 1 до  $N$ , другие концы проводов нумеруются от  $N + 1$  до  $2N$ . Номера узлов лежат в диапазоне от  $2N + 1$  до  $2N + M$ .

Следующие  $K$  строк содержат описания связей между концами проводов и узлами:  $i$ -я строка содержит два целых числа  $a_i$ ,  $b_i$  – номера связанных концов проводов или узлов ( $1 \leq i \leq K$ ,  $1 \leq a_i, b_i \leq 2N + M$ ). Гарантируется, что каждый конец провода связан напрямую только с одним узлом или другим концом провода. Связь между двумя узлами подразумевает под собой то, что между этими узлами проходит один или несколько проводов.

### Формат выходных данных

В выходной файл нужно вывести одно целое число – минимальное количество проверок, которое нужно сделать Коле.

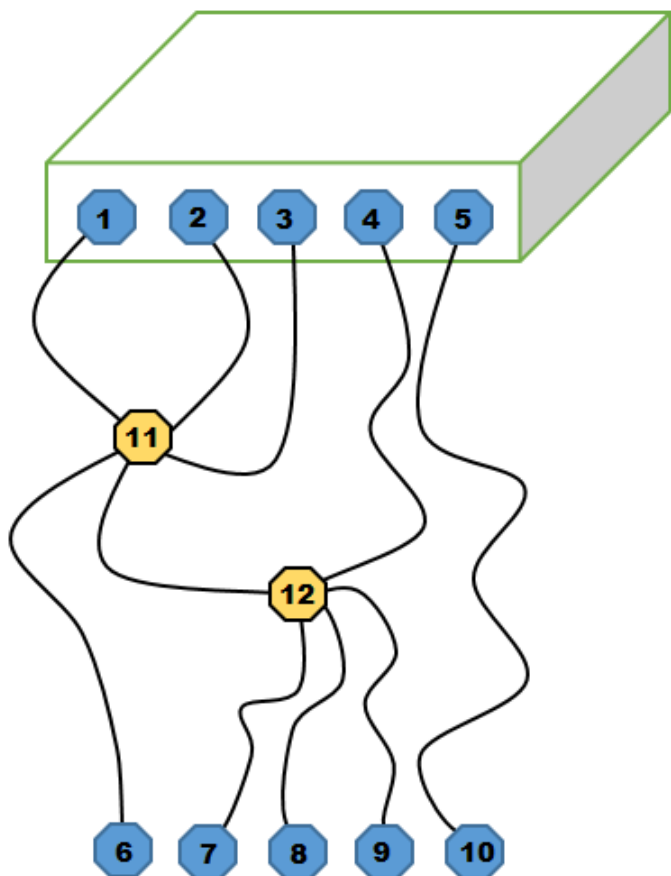
### Примеры

input.txt	output.txt
2 1 4 1 5 2 5 5 3 5 4	1
5 2 10 1 11 2 11 3 11 11 6 11 12 4 12 12 7 12 8 12 9 5 10	6

## Замечание

В первом тестовом примере всего два провода, на которых завязался один узел. Для того, чтобы однозначно понять какой провод соответствует какому устройству достаточно сделать одну проверку. Даже если она будет неудачной, и устройство не заработает, Коля поймет, что этому проводу соответствует другое устройство, а для оставшегося провода все будет однозначно.

На рисунке ниже показано расположение проводов во втором тестовом примере.



## Задача D. Контрольная по алгебре

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Фёдор Михайлович, учитель математики в специализированном учебно-научном центре города Диманово, готовит контрольную на тему «делимость целых чисел». Времени осталось мало, а различных вариантов контрольной нужно придумать много, ведь ученики, решающие одинаковый вариант, могут списать решения друг у друга!

Фёдор Михайлович придумал следующую задачу: пусть даны целые числа  $A$  и  $N$ , найдите целое число  $B$  такое, что  $AB + A + B$  делится на  $N$ . Он решил сделать несколько похожих задач, меняя данные в условии задачи. Естественно, ему нужно знать и ответы на свои задачи. Помогите Фёдору Михайловичу написать программу, которая будет решать его задачу для разных исходных данных, сэкономит много времени и сил при подготовке контрольной работы.

### Формат входных данных

Единственная строка входного файла содержит два целых числа  $A$ ,  $N$ , разделённых знаком пробела – числа, которые будут использоваться для очередного варианта контрольной ( $1 \leq A, N \leq 10^9$ ).

### Формат выходных данных

В выходной файл нужно вывести единственное число  $B$  – ответ на задачу Фёдора Михайловича. Число  $B$  должно быть неотрицательным и не превышать  $10^9$ . Если такого  $B$  не найдется – необходимо вывести число  $-1$ .

### Примеры

<code>input.txt</code>	<code>output.txt</code>
2 4	2
3 4	-1

## Задача Е. Дедлайн Макара

Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Сегодня у Макара дедлайн на работе по проекту. Нужно СРОЧНО написать недостающий модуль, необходимый для запуска релиза программы. Но вот незадача – написать его нужно на особом языке `макаг8`, известном только в узких кругах. Поэтому Макара обратился к Вам за помощью. Но как же написать код, если неизвестен даже синтаксис языка?

За этим вопросом Вы обратились в интернет и нашли код нужного модуля на сайте `stackoverflow.com`. С этой радостной новостью Вы тут же поделились с Макаром, но он не воспринял её с должным энтузиазмом. «Код со `stackoverflow` копируют только неумехи», — сказал он.

Поэтому Вы пошли на хитрость, предложив придумать код вместе. Идея хитрости заключается в том, чтобы Макара сам придумал код, написанный на `stackoverflow`.

Вы уже знаете программу. Предположим, что она записана в строку  $S$ . В процессе вашего диалога Макару в голову приходят блестящие мысли в виде кусков кода. Любую очередную блестящую мысль можно либо проигнорировать, либо дописать в конец составляемой строки какой-то начальный фрагмент этой блестящей мысли. Начальный фрагмент – это последовательный набор символов, который идёт от начала строки. Его нужно записать так, чтобы он посимвольно совпал со следующей некоторой частью строки  $S$ , которую вы строите.

Например, если вам нужно построить строку `"abcdeshka"`, для которой вы уже вышеописанным способом составили `"ab"`, а Макару пришла очередная блестящая мысль `"cdrom"`, то Вы можете либо отбросить эту мысль, либо добавить `"c"` или `"cd"` к `"ab"`, тем самым получив `"abc"` или `"abcd"` соответственно.

При этом всё Вы стремитесь использовать как можно меньше идей Макара, то есть хотите проигнорировать максимальное количество его блестящих мыслей. Гарантируется, что составить программу из мыслей Макара можно.

Помогите Макару «придумать», известную уже Вам программу  $S$ .

### Формат входных данных

В первой строке входного файла записано единственное целое число  $N$  ( $1 \leq N \leq 300$ ).

Вторая строка содержит символы строки  $S$  – текст нужной программы длины  $N$  (строка состоит только из символов строчных латинских букв `'a' – 'z'`).

В третьей строке дано целое число  $K$  – количество блестящих мыслей Макара ( $1 \leq K \leq 300$ ).

Следующие  $K$  строк содержат описание блестящих мыслей Макара. Каждая строка соответствует ровно одной блестящей мысли и состоит только из строчных латинских букв. Гарантируется, что суммарная длина мыслей не превышает 300.

### Формат выходных данных

В выходной файл нужно вывести одно целое число  $M$  – максимальное количество проигнорированных блестящих мыслей Макара при написании программы.

## Примеры

input.txt	output.txt
9 abcdeshka 6 ab cdrom e es esh ka	2
5 aaaaa 5 a aa aaa aaaa aaaaa	4